UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche

N° 1305

Programme 3
Réseaux et Systèmes Répartis

A STUDY ON THE CACHE MEMORY MISS RATIO ISSUE IN MULTIPROCESSOR SYSTEMS

Ken CHEN

Octobre 1990

A Study on the Cache Memory Miss Ratio Issue in Multiprocessor Systems

Ken CHEN*

Projets MEVAL / REFLECS, INRIA, B.P. 105
F-78153 Le Chesnay Cedex, FRANCE
Telephone: +(33)-1-39 63 52 84
E-mail: kc@score.inria.fr

September 21, 1990

Abstract

This paper addresses the problem of evaluating the miss ratio of cache memory in multiprocessor shared-memory shared-channel systems. Whereas the read/write nature of the references access have no impact on the cache memory miss ratio of a monoprocessor system, it does have an impact on the multiprocessor shared-memory shared-channel systems, due to cache coherence control. In this paper, we consider a system consisting of two identical processors, each having a fully associative snooping cache memory, managed by the LRU policy. These two processors are connected to the main shared-memory via a shared-channel (bus). We have derived exact expressions of miss ratio, for the *local* missing (limited to one's own cache) as well as for the *global* missing (cache set vs. main memory). The essential part of this work is based on the working set size evaluation.

Keywords: Cache memory, Working set, Miss ratio, Multiprocessor system

Une Etude sur le Défaut de Cache dans les Systèmes Multiprocesseurs

Résumé

Ce papier aborde le problème de l'évaluation du défaut de cache dans les systèmes multiprocesseurs avec mémoires et canal partagés. Alors que la nature des accès (lecture/écriture) aux références n'a pas d'impact sur le défaut de cache dans un système monoprocesseur, elle doit cependant être prise en compte dans un système multiprocesseur avec mémoires et canal partagés, à cause du maintien de la cohérence des mémoires. Dans ce papier, nous considérons un système avec deux processeurs identiques, chacun d'eux possède un *fully associative snooping cache memory* géré selon la politique LRU. Ces deux processeurs sont reliés à la mémoire centrale par un bus. Nous distinguons deux types de défaut de cache: *local* (défaut limité à son propre cache) et *global* (défaut dans l'ensemble des caches). Basé sur l'évaluation de la taille de l'ensemble de travail (*working set*), nous avons obtenu les expressions exactes pour ces deux types de défaut de cache.

Mots Clés: Mémoires caches, Ensemble de travail, Défaut de cache, Systèmes multiprocesseurs

*boursier post-doctoral de l'INRIA. (Supported by a post-doctoral scholarship of INRIA.)

A Study on the Cache Memory Miss Ratio Issue in Multiprocessor Systems

Ken CHEN

Projets MEVAL / REFLECS, INRIA, B.P. 105
F-78153 Le Chesnay Cedex, FRANCE
Telephone: +(33)-1-39 63 52 84
E-mail: kc@score.inria.fr

September 21, 1990

Abstract

This paper addresses the problem of evaluating the miss ratio of cache memory in multiprocessor shared-memory shared-channel systems. Whereas the read/write nature of the references access have no impact on the cache memory miss ratio of a monoprocessor system, it does have an impact on the multiprocessor shared-memory shared-channel systems, due to cache coherence control. In this paper, we consider a system consisting of two identical processors, each having a fully associative snooping cache memory, managed by the LRU policy. These two processors are connected to the main shared-memory via a shared-channel (bus). We have derived exact expressions of miss ratio, for the *local* missing (limited to one's own cache) as well as for the *global* missing (cache set vs. main memory). The essential part of this work is based on the working set size evaluation.

Keywords: Cache memory, Working set, Miss ratio, Multiprocessor system

1 Introduction

As we are witnessing since the recent years, cache memories are becoming a basic computer system component, not only for mainframes but also for high speed microprocessor based personal computers. Caches are small, high speed buffers used to hold a portion of the low speed main memory, in order to reduce the waiting time fraction of the CPU (Central Processing Unit). This two level storage scheme began to be used about 20 years ago under the (main memory, storage device) context [Den68], then the advances in technology switched the context to (cache, main memory)[Smi82]. One of the current trends in computer architecture is towards parallel systems. With cache implementation in multiprocessor shared-memory environments, some new problems arise, which do not exist in a monoprocessor system, in

particular the cache coherence control. This paper is concerned mainly with this issue.

Hereafter, we recall briefly some basic notions relative to the cache memories (an excellent survey on cache memories can be found in [Smi82]):

The basic unit of a cache memory is line, which contains R main memory references. Internally, a cache is divided into S segments, each segment contains L lines. So, the organization of a cache can be described by the triple (S, L, R). We have different types of cache organization, *e.g.* fully associative, direct mapped, set associative, etc. A fully associative cache is characterized by $S = 1$.

In equilibrium, when a missing reference is fetched into cache, it takes the place occupied by another reference, consequently a replacement algorithm is required. The basic replacement algorithms are First In First Out (FIFO), Last In First Out (LIFO), Random Replacement(RR) and Least Recently Used (LRU), of which LRU is in general the most efficient. LRU consists in replacing the least recently used reference by the newly fetched one.

Program behavior is also an important factor. Caching mechanism owns its efficiency to the observed fact (*e.g.* from trace-driven studies) that programs enjoy the properties of *spacial* and *temporal* localities, *i.e.* over a short period, some contiguous references will be accessed more frequently than others. A popular program behavior model is the independent reference model (IRM) [Rao78]. Although IRM reflects only partially the spacial and temporal localities, this model is the most widely used, since on one hand it's mathematically tractable, and on the other hand it yields a rather accurate insight of the performances.

A commonly used metric for cache performance evaluation is the *miss ratio*, defined as the number of main memory accesses over the total number of accesses. This is a very important parameter, since it can be used to evaluate the CPU waiting time, the data transfer rate between the main memory and cache, etc. In this paper, we propose an approach for the evaluation of miss ratio in multiprocessor systems.

In a monoprocessor system, the miss ratio depends basically on the program behavior, the cache organization and the replacement algorithm. It is insensitive of the read/write nature of the instructions.

In a multiprocessor system, things become more complex. There are a certain number of possible architectures for multiprocessor systems, *e.g.* hypercube, shared-channel (bus). In this paper, we restrict our investigation on cached multiprocessor bus-linked shared-memory systems, *i.e.* a set of processors having each a local cache memory and connected to a shared main memory via a bus. A line of references can be fetched from the main memory, but also from another cache. In general, caches are linked together by a high speed bus, compared to the link between the caches and main memory. For such a system, it is possible that a given main shared-memory reference be located in multiple caches, thus the problem of cache coherence arises. Although it is possible to prevent this multicopy situation by some software approaches [AB86], these approaches are in general not efficient; a very popular approach is to allow all references to be cached by all processors, and to implement a cache coherence protocol to insure data consistency.

One of the particularities of a bus-type link is that the traffic (transaction) is broadcasted. Based on it, different coherence protocols, known as *snooping tech-*

nique, have been proposed and/or implemented [AB86, GMR87]. Currently the snooping technique is the most used, its principle can be summarized as following: Each cache has a controller which observes the bus. Each cache line has a local status indicator {Valid, Invalid}. Each time a line of reference is cached in, it is marked *valid*. All *write* actions are broadcasted. So, whenever a processor writes to a reference, all the lines in other caches with contains a copy of the reference, if any, are marked *invalid* by the local controllers. There are two ways for updating, namely *Write Once* and *Write Through*. In *Write Once* scheme, the holder of the valid reference write its current value back to the main memory when the reference is to be moved out of caches; whereas in *Write Through* scheme, the write back takes place at each modification. Thus, at any time, the good version of a reference is located in one cache and/or in the main memory. An invalidated reference physically still occupies a cache location, but the line containing it is logically missing. Upon request (*read* or *write*) of a reference, if this one is missing (physically or logically), the controller will first try to get it from the cache which holds the valid version, if any; otherwise, the fetch will be done from the main memory.

From this short introduction to the cache implementation in shared-bus shared memory multiprocessor system, it is obvious that the read/write nature of the instructions has to be taken into consideration for the evaluation of miss ratio. It appears also that there are two kinds of misses for a given processor: i) *local*, *i.e.* the reference is missing in its local cache; and ii) *global*, *i.e.* the reference is missing in all caches. From the system designer's point of view, the former helps to determine the amount of transactions between caches; and the later helps to determine the amount of transactions between the set of caches and the main memory.

There has been a considerable number of papers addressing the cache architecture, algorithms, program behavior and performance evaluation [Rao78, Smi82, AB86, ACK87, FGT87, GMR87, Goo87, AHH89, FR90]. Whereas miss ratio is often used as a basic parameter, no much attentions have been given to the evaluation of the miss ratio itself in multiprocessor systems.

In this paper, we evaluate the miss ratio of a system having two identical processors, each having a cache. Section 2 gives a general formulation, and identifies the basic problem as the evaluation of the working set of each cache. The problem having a combinatorial nature, in section 3, we have derived exact expressions for some simplified models. The interest of this study is that the simplified models give a reasonably good representation of the caching behavior in multiprocessor systems, numerical results are produced in section 4.

2 Modelling

2.1 Conditions and notations

We consider a couple of identical processors (**X**, **Y**) with caches (K_X , K_Y). The caches are fully associative ($S = 1$), and are managed using the LRU policy, their size are C lines.

For the sake of simplicity, z (resp. Z) will refers to either x or y (resp. X or

Y), and $\overline{z}$ (resp. $\overline{Z}$) the remainning one, without explicit indication. The reference stream received by the processor **Z** is denoted by Z= $\{z_t\}_{t=0..\infty}$ ($z \in \{x,y\}$, $Z \in \{X,Y\}$). The references are taken into account in cache line level, *i.e.* the word *reference* is to be understood subsequently as a line of physical references.

X and **Y** execute two different programs sharing the same references set, denoted by $\mathcal{R}$. Let $N = Card(\mathcal{R})$ be the total number of reference lines, we consider only the cases $N \geq 2C$. The program behavior model is IRM, *i.e.* the probability of accessing a given reference r issued from $\mathcal{Z}$ is $p_{z,r}$, with $\sum_{r=1}^{N} p_{z,r} = 1$ and $\forall r, p_{z,r} > 0$. For simplicity, we define also $q_{z,r} := 1 - p_{z,r}$.

We suppose further that read and write instructions are randomly associated to any reference r, with write instruction occurrence probability (write ratio) $w_{z,r}$, $w_{z,r} \in [0..1]$, *i.e.* the probability of accessing r from **Z** with a write instruction is $w_{z,r}p_{z,r}$, which will be denoted by $p^w_{z,r}$; parallely, we have $q^w_{z,r} = 1 - p^w_{z,r}$.

When a reference is accessed with a write instruction, the same reference located in another cache, if any, is marked *invalid*. The only ambiguity arises when the same reference is accessed by both processors with write instruction during the same cycle, we assume that this conflict is solved such that K_X has the valid reference, and K_Y has the invalid one.

2.2 General case

We are interested in the local and global miss ratio, or dually, in

1. P_Z: the probability of having the current reference from $\mathcal{Z}$ valid in K_Z
2. $P_{Z\overline{Z}}$: the one of finding the reference from $\mathcal{Z}$ valid in one or another cache.

Under the fully associativity, $N \geq 2C$, $p_{z,r} > 0$ and LRU conditions, it has been proved [Rao78] that P_Z and $P_{Z\overline{Z}}$ can be expressed as:

$$P_Z = \sum_{r=1}^{N} p_{z,r} P_{Z,r} \quad (1)$$

$$P_{Z\overline{Z}} = \sum_{r=1}^{N} p_{z,r}(P_{X,r} + P_{Y,r} - P_{xy,r}) \quad (2)$$

where $P_{Z,r}$ is the probability of having the reference r in K_Z and valid, and $P_{xy,r}$ the one of having r valid in K_X and K_Y. Consequently, it is sufficient to find the expressions of $P_{Z,r}$ and $P_{xy,r}$.

Assuming the system in equilibrium, denote by $R_{z,l,r}$ the probability of having r in K_Z at the current instant t, either valid or invalid, knowing that the last access to this reference occurred l time units before. $R_{z,l,r}$ is simply:

$$R_{z,l,r} = Prob(Card(W_Z[t - l + 1,\ t - 1]) < C)$$

where $W_Z[t_1, t_2]$ is the *working set* between $[t_1, t_2]$, *i.e.* the set of references occurring in $\mathcal{Z}$ between $[t_1, t_2]$: $W_Z[t_1, t_2] = \{r / \exists t \in [t_1, t_2], z_t = r\}$ [Den68, Len74].

For $P_{X,r}$, we must distinguish two cases, a) the last write access from $\mathcal{Y}$ on r occurred i time before, without simultaneously a write instruction on r from $\mathcal{X}$; b) the two accesses occurred at the same time. This gives:

$$P_{X,r} = p^w_{x,r}\, p^w_{y,r} \sum_{i=1}^{\infty} (q_{x,r} q_{y,r})^{i-1} R_{x,i,r} + q^w_{y,r}\, p_{x,r} \{ \sum_{i=1}^{\infty} [q_{x,r}\, q^w_{y,r}]^{i-1} R_{x,i,r} \}$$

A similar reasoning gives $P_{Y,r}$, which is slightly different from $P_{X,r}$, due to the write precedence rule:

$$P_{Y,r} = q^w_{x,r} p_{y,r} \{ \sum_{j=1}^{\infty} [q_{y,r} p^w_{x,r}]^{j-1} R_{y,j,r} \}$$

and the one of $P_{xy,r}$:

$$\begin{aligned} P_{xy,r} &= (1 - w_{x,r})(1 - w_{y,r}) p_{x,r}\, p_{y,r} [\sum_{i=1}^{\infty} [q_{x,r}{}^{i-1} R_{x,i,r}][q_{y,r}{}^{j-1} R_{y,j,r}] \\ &+ (1 - w_{y,r}) p_{x,r}\, p_{y,r} \sum_{i=2}^{\infty} [q_{x,r} q^w_{y,r}]^{i-1} R_{x,i,r} \{ \sum_{j=1}^{i-1} (\frac{q_{y,r}}{q^w_{y,r}})^{j-1} R_{y,j,r} \} \\ &+ (1 - w_{x,r}) p_{x,r} p_{y,r} \sum_{j=2}^{\infty} [q_{y,r} q^w_{x,r}]^{j-1} R_{y,j,r} \{ \sum_{i=1}^{j-1} (\frac{q_{x,r}}{q^w_{x,r}})^{i-1} R_{x,i,r} \} \end{aligned}$$

Now the solution depends basically on the expressions of $R_{z,l,r}$, whose computation is of combinatorial nature. As a matter of fact, at each instant within $[t - l + 1,\ t - 1]$, every reference may occur. Such a problem is rather complex in general, but has a solution if all references occur equiprobably. A typical case is the urn problem [Fel68, pp. 101-103], which will be the basis of the rest of this paper.

In a first step, we derive the expressions of $P_{Z,r}$ and $P_{xy,r}$ for the case where all the references have the same probability (referenced subsequently as *e-distribution*). From this basic expression, we will extent the study to the case where the references are partitionned into two subsets in which they have the same access probability (referred subsequently as *b-distribution*). This modelling has already some practical interests. Indeed, on the one hand, in parallel computing, often several processors operate on a same data bloc (public part) whereas each processor has its own data bloc (private part), *e.g.* two programs sharing a common array and doing different computations, the b-distribution model will be suitable for these situations. On the other hand, from the computational point of view, although we can theoretically derive some multi-subset partitions models, they would require much more computation time than the one required by the b-distribution model, since the problem is inherently complex. Thus, the b-distribution model seems to be a good tradeoff between accuracy and complexity.

2.3 The basic expression

2.3.1 The urn problem

The working set size problem can be transformed into the urn (random ball distribution) problem. First, we recall the urn problem:

Following the notations used in [Fel68], $P_m(r,n)$, the probability of having exactly m cells empty over n cells, after a random distribution with equal distribution probability n^{-r} of r balls is:

$$P_m(r,n) = C_n^m \sum_{v=0}^{n-m} (-1)^v C_{(n-m)}^v (1 - \frac{m+v}{n})^r$$

Thus, the probability of having exactly l cells occupied, $p_l^o(r,n)$, is:

$$p_l^o(r,n) = C_n^l \sum_{v=0}^{l} (-1)^v C_l^v (\frac{l-v}{n})^r$$

And the probability of having less than (not including) L cells occupied, $P_L^o(r,n)$, is

$$P_L^o(r,n) = \sum_{l=0}^{L-1} p_l^o(r,n) = \sum_{l=0}^{L-1} C_n^l \sum_{v=0}^{l} (-1)^v C_l^v (\frac{l-v}{n})^r$$

Now return to our working set size problem, since the remaining $N-1$ references occur equiprobably with probability $\frac{1}{N-1}$, thus $R_{z,i} = P_{C-1}^o(i-1, N-1)$, or explicitly:

$$R_{z,i} = \sum_{l=0}^{C-1} C_{N-1}^l \sum_{v=0}^{l} (-1)^v C_l^v (\frac{l-v}{N-1})^{i-1}$$

For the sake of simplicity, we define the operator $\mathcal{S}_{l,v}[f]$, where f is a function of (l,v), as:

$$\mathcal{S}_{l,v}[f] = \sum_{l=0}^{C-1} C_{N-1}^l \sum_{v=0}^{l} (-1)^v C_l^v f$$

Then $R_{z,i}$ can be expressed as:

$$R_{z,i} = \mathcal{S}_{l,v}[(\frac{l-v}{N-1})^{i-1}]$$

Remark: The operator $\mathcal{S}_{l,v}$ and $\frac{l-v}{N-1}$ are both independent of i, so we have:

$$\sum_i \mathcal{S}_{l,v}[(\frac{l-v}{N-1})^{i-1}] = \mathcal{S}_{l,v}[\sum_i (\frac{l-v}{N-1})^{i-1}]$$

2.3.2 Z e-distributed

When all the reference access probabilities are equal in stream $\mathcal{Z}$, *i.e.* $\forall r \in [1..N]\ p_{z,r} = 1/N$, $R_{z,i,r}$ is simply $R_{z,i}$. After some straightforward computations, the expression of $P_{X,r}$ for X e-distributed is found to be:

$$P_{X,r} = \mathcal{S}_{l,v}[\frac{w_{x,r}p^w_{y,r}}{N - q_{y,r}(l-v)} + \frac{q^w_{y,r}}{N - q^w_{y,r}(l-v)}]$$

As $\mathcal{S}_{l,v}$ is independent of the distributions of $w_{z,r}$, and $p_{y,r}$, we have:

$$P_X = \mathcal{S}_{l,v}[\sum_{r=1}^{N}\{\frac{w_{x,r}p^w_{y,r}}{N - q_{y,r}(l-v)} + \frac{q^w_{y,r}}{N - q^w_{y,r}(l-v)}\}]$$

whose value depends on the distributions of the write ratio $\{w_{z,r}\}_{r=1..N}$ and the distribution $\{p_{y,r}\}_{r=1..N}$

Similarly, if Y is e-distributed, we have:

$$P_Y = \mathcal{S}_{l,v}[\sum_{r=1}^{N}\{\frac{q^w_{x,r}}{N - q^w_{x,r}(l-v)}\}]$$

2.3.3 X and Y e-distributed

When X and Y are both e-distributed, we can derive the expression of $P_{xy,r}$, so give a complete solution to the problem. After some calculus:

$$\begin{aligned} P_{xy,r} &= \mathcal{S}_{l,v}[\mathcal{S}_{l',v'}[\frac{(1-w_{x,r})(1-w_{y,r})N^2}{N^2-(l-v)(l'-v')}]] \\ &+ \sum_{z \in \{x,y\}} \mathcal{S}_{l,v}[\mathcal{S}_{l',v'}[(\frac{(1-w_{z,r})(N-w_{z,r})}{N^2[N-w_{z,r}-(l-v)]}) \\ &\times(\frac{(N-w_{z,r})(l'-v')}{N^2-(N-w_{z,r})(l'-v')} - \frac{(l-v)(l'-v')}{N^2-(l-v)(l'-v')})]] \end{aligned}$$

Now, according to the equations ?? and ??, the quantities P_Z and $P_{Z\overline{Z}}$ depend only on the distribution $\{w_{z,r}\}_{r=1..N}$.

2.4 B-distributed case

In the b-distributed case, for each stream $\mathcal{Z}$, we have two distinct reference subsets $\mathcal{A}_z$ and $\mathcal{B}_z$, with $\mathcal{A}_z \cup \mathcal{B}_z = \mathcal{R}$ and $\mathcal{A}_z \cap \mathcal{B}_z = \emptyset$. Let $N_{z,a} = card(\mathcal{A}_z)$, and let a_z and b_z be such that $a_z + b_z = 1$ and $a_z b_z > 0$. For $r \in \mathcal{A}_z$, by definition, $p_{z,r} = \frac{a_z}{N_{z,a}}$, we denote it by $p_{z,a}$, parallely $q_{z,a} = 1 - p_{z,a}$, in addition $R_{z,i,r}$ can be simplified to $R_{z,i,a}$. Similarly, we define $N_{z,b}$, $p_{z,b}$, $q_{z,b}$ and $R_{z,i,b}$.

2.4.1 Expressions of $R_{z,i,a}$ and $R_{z,i,b}$

Locally, we have both e-distribution cases in $\mathcal{A}_z$ and $\mathcal{B}_z$. After some computations, we have found:

$$R_{z,i,a} = \mathcal{T}_{z,a}[(\frac{U_z}{q_{z,a}})^{i-1}] \quad \text{and} \quad R_{z,i,b} = \mathcal{T}_{z,b}[(\frac{U_z}{q_{z,b}})^{i-1}]$$

where U_z is a function of (l_a, l_b, v_a, v_b) given by:

$$U_z = p_{z,a}(l_a - v_a) + p_{z,b}(l_b - v_b)$$

and $\mathcal{T}_{z,a}$, $\mathcal{T}_{z,b}$ are operators defined by:

$$\mathcal{T}_{z,a}[f] = \sum_{l_a=0}^{\min(C^*, N^*_{z,a})} \sum_{l_b=0}^{\min(C^*-l_a, N_{z,b})} C^{l_a}_{N^*_{z,a}} C^{l_b}_{N_{z,b}} \sum_{v_a=0}^{l_a} \sum_{v_b=0}^{l_b} (-1)^{v_a+v_b} C^{v_a}_{l_a} C^{v_b}_{l_b} f$$

$$\mathcal{T}_{z,b}[f] = \sum_{l_b=0}^{\min(C^*, N^*_{z,b})} \sum_{l_a=0}^{\min(C^*-l_b, N_{z,a})} C^{l_b}_{N^*_{z,b}} C^{l_a}_{N_{z,a}} \sum_{v_b=0}^{l_b} \sum_{v_a=0}^{l_a} (-1)^{v_a+v_b} C^{v_a}_{l_a} C^{v_b}_{l_b} f$$

in which f is a function of (l_a, l_b, v_a, v_b), and $C^* = C - 1$ (idem for $N^*_{z,a}$ and $N^*_{z,b}$). Similarly to the e-distribution case, we have:

$$\sum_i \mathcal{T}_{z,a}[(\frac{U_z}{q_{z,a}})^{i-1}] = \mathcal{T}_{z,a}[\sum_i (\frac{U_z}{q_{z,a}})^{i-1}]$$

idem for $\mathcal{T}_{z,b}$.

2.4.2 $\mathcal{Z}$ b-distributed

When $\mathcal{X}$ is b-distributed, *i.e.* $p_{x,r} = \frac{a_x}{N_{x,a}}$, for $r \in \mathcal{A}_x$ and $p_{x,r} = \frac{b_x}{N_{x,b}}$, for $r \in \mathcal{B}_z$. Denote $P_{X,r}$ by $P_{Xa,r}$, for $r \in \mathcal{A}_x$; idem for $P_{Xb,r}$, we have:

$$P_{Xa,r} = \mathcal{T}_{x,a}[p_{x,a}(\frac{w_{x,r}p^w_{y,r}}{1 - q_{y,r}U_x} + \frac{q^w_{y,r}}{1 - q^w_{y,r}U_x})]$$

$$P_{Xb,r} = \mathcal{T}_{x,b}[p_{x,b}(\frac{w_{x,r}p^w_{y,r}}{1 - q_{y,r}U_x} + \frac{q^w_{y,r}}{1 - q^w_{y,r}U_x})]$$

Similarly, when $\mathcal{Y}$ is b-distributed, we have:

$$P_{Ya,r} = \mathcal{T}_{y,a}[p_{y,a}(\frac{q^w_{x,r}}{1 - q^w_{x,r}U_y})]$$

$$P_{Yb,r} = \mathcal{T}_{y,b}[p_{y,b}(\frac{q^w_{x,r}}{1 - q^w_{x,r}U_y})]$$

2.4.3 $\mathcal{X}$ and $\mathcal{Y}$ b-distributed

When $\mathcal{X}$ and $\mathcal{Y}$ are both b-distributed, for a given reference r, we have exactly four possibilities:

1. $r \in \mathcal{A}_x \cap \mathcal{A}_y = \mathcal{R}_{aa}$, (We denote $N_{aa} = Card(\mathcal{R}_{aa})$).
2. $r \in \mathcal{A}_x \cap \mathcal{B}_y = \mathcal{R}_{ab}$, (We denote $N_{ab} = Card(\mathcal{R}_{ab})$).
3. $r \in \mathcal{B}_x \cap \mathcal{A}_y = \mathcal{R}_{ba}$, (We denote $N_{ba} = Card(\mathcal{R}_{ba})$).
4. $r \in \mathcal{B}_x \cap \mathcal{B}_y = \mathcal{R}_{bb}$, (We denote $N_{bb} = Card(\mathcal{R}_{bb})$).

Computationally speaking, only the index changes. Denoting $P_{xy,r}$ by $P_{cd,r}$ for $r \in \mathcal{R}_{cd}$, $(c,d) \in \{a,b\}^2$, we get:

$$\begin{aligned} P_{cd,r} &= p_{x,c}p_{y,d}\mathcal{T}_{x,c}[\mathcal{T}_{y,d}[\{\frac{(1-w_{x,r})(1-w_{y,r})}{(1-U_xU_y)} \\ &+ (\frac{(1-w_{y,r})q^w_{y,d}}{1-q^w_{y,d}-U_y})\{\frac{q^w_{y,d}U_x}{1-q^w_{y,d}U_x} - \frac{U_xU_y}{1-U_xU_y}\} \\ &+ (\frac{(1-w_{x,r})q^w_{x,c}}{1-q^w_{x,c}-U_x})\{\frac{q^w_{x,c}U_y}{1-q^w_{x,c}U_y} - \frac{U_xU_y}{1-U_xU_y}\} \ \}]] \end{aligned}$$

2.4.4 Uniform subsets case

A simple yet interesting case is the following: the write ratio is the same for references issuing from the same subset (four in total), *i.e.* $w_{z,r} = w_{z,e}$ for $z \in \{x,y\}$ and $e \in \{a,b\}$. In this case, the expressions are:

$$\begin{aligned} P_X &= \sum_{c\in\{a,b\}}\sum_{d\in\{a,b\}}\sum_{r\in\mathcal{R}_{cd}} p_{x,r}P_{Xc,r} \\ &= \sum_{c\in\{a,b\}}\sum_{d\in\{a,b\}} N_{cd}\mathcal{T}_{x,c}[{p_{x,c}}^2(\frac{w_{x,c}w_{y,d}p_{y,d}}{1-q_{y,d}U_x} + \frac{1-w_{y,d}p_{y,d}}{1-(1-w_{y,d}p_{y,d})U_x})] \\ P_Y &= \sum_{c\in\{a,b\}}\sum_{d\in\{a,b\}}\sum_{r\in\mathcal{R}_{cd}} p_{y,r}P_{Yd,r} \\ &= \sum_{c\in\{a,b\}}\sum_{d\in\{a,b\}} N_{cd}\mathcal{T}_{y,d}[{p_{y,d}}^2(\frac{1-w_{x,c}p_{x,c}}{1-(1-w_{x,c}p_{x,c})U_y})] \\ P_{XY} &= \sum_{c\in\{a,b\}}\sum_{d\in\{a,b\}}\sum_{r\in\mathcal{R}_{cd}} p_{x,r}(P_{Xc,r}+P_{Yd,r}-P_{cd,r}) \\ &= \sum_{c\in\{a,b\}}\sum_{d\in\{a,b\}} N_{cd}\mathcal{T}_{x,c}[{p_{x,c}}^2(\frac{w_{x,c}w_{y,d}p_{y,d}}{1-q_{y,d}U_x} + \frac{1-w_{y,d}p_{y,d}}{1-(1-w_{y,d}p_{y,d})U_x})] \\ &+ \sum_{c\in\{a,b\}}\sum_{d\in\{a,b\}} N_{cd}\mathcal{T}_{y,d}[p_{x,c}p_{y,d}(\frac{1-w_{x,c}p_{x,c}}{1-(1-w_{x,c}p_{x,c})U_y})] \end{aligned}$$

$$+ \sum_{c \in \{a,b\}} \sum_{d \in \{a,b\}} N_{cd} {p_{x,c}}^2 p_{y,d} \mathcal{T}_{x,c} [\mathcal{T}_{y,d} [\{ \frac{(1-w_{x,r})(1-w_{y,r})}{(1-U_x U_y)}$$

$$+ (\frac{(1-w_{y,r})(1-w_{y,r}p_{y,d})}{1-w_{y,r}p_{y,d}-U_y})\{\frac{(1-w_{y,r}p_{y,d})U_x}{1-(1-w_{y,r}p_{y,d})U_x} - \frac{U_x U_y}{1-U_x U_y}\}$$

$$+ (\frac{(1-w_{x,r})(1-w_{x,r}p_{x,c})}{1-w_{x,r}p_{x,c}-U_x})\{\frac{(1-w_{x,r}p_{x,c})U_y}{1-(1-w_{x,r}p_{x,c})U_y} - \frac{U_x U_y}{1-U_x U_y}\}$$

$$\}]]$$

And for P_{YX} the symmetrical expression.

These expressions can be programmed for numerical computation without difficulties.

3 Numerical results

The case we present here has cache size $C = 5$ reference lines, and reference set size $N = 100$ reference lines, the limitation in cache size is due to computational difficulties. We show the local and global presence probability for **X**, to which the ones of **Y** are very close. In the first two figures, the write ratio is the same for all references, taken as 0.25, which is a commonly observed value in trace-driven studies. The numerical results are computed using the symbolic computation language *MAPLE*.

First, we set N_{ab}=N_{ba}=0, *i.e.* the two reference subsets partitions are the same. A corresponding real case could be the parallel execution of two programs manipulating two data blocs ($\mathcal{R}_{aa}$ and $\mathcal{R}_{bb}$). Curves for various sizes (N_{aa}=10 (curve A) till N_{aa}=90 (curve E)) have been presented, for various value of relative per-reference access frequency (p_a/p_b) (cf. Figure 1).

The worst case corresponds to the one in which references in the two blocs are equiprobably accessed. (We get a degenerated one bloc situation).

One can also observe that a tradeoff exists between the size of a bloc and its relative access frequency, *e.g.* for $p_a/p_b > 1$, the curve B, which is an intermediary case, is above the curves A and E, the two extreme cases.

We then set $N_{ab} = N_{ba} > 0$ and $N_{aa} = N_{bb} > 0$, *i.e.* there is mixture of the reference subsets of the two streams. A corresponding real case could be the parallel manipulation on the same array, with different algorithms. Curves for various size of these blocs (N_{aa}=5 (curve A) till N_{aa}=45 (curve E)), for various values of relative per-reference access frequency (p_a/p_b) (cf. Figure 2). Once more, we get the worst case when accesses are symmetrically distributed. And, as expected intuitively, the miss ratio increases when the degree of mixture increases, in particular, the worst case (A) has the biggest mixed segments with $N_{ab} = N_{ba} = 45$.

We also examined the write ratio impact. In this study, we set N_{ab}=N_{ba}=0, and we draw curves for write ratio varying from 0 (curve A) to 1 (curve E), when N_{aa} varying from 10 to 90 (cf. Figure 3). These curves show the weak impact of write ratio in local cache miss ratio, whereas this impact is almost negligible for global cache missing.

4 Conclusion

This work is focused on the evaluation of the local and global cache miss ratio in a multiprocessor shared-channel shared-memory system consisting of two identical processors with snooping cache memories, under the conditions of fully associative cache organization, LRU replacement algorithm and IRM program behavior.

At the first step, we have derived exact expressions for miss ratio by assuming that references are accessed equiprobably. This basic expression has been extended to the case where each processor considers the reference set as two distinct subset of equiprobably accessed references. Despite this strong approximation, many scenarii can be evaluated by varying: a) reference set/subset sizes; b) cache size; and c) write ratio distributions. A computational example has been presented.

Using the same technique, an extension to more than two sets is possible, but the computation amount will increase greatly, since the problem has inherently a high complexity.

In the current formulation, cache sizes are assumed to be equal. We can easily get a formulation with K_X and K_Y having different size. However, we do not consider this case, since it is not common in multiprocessor system architectures.

Acknowledgements

The author would like to thank Dr. Ph. ROBERT for his numerous helps and suggestions.

References

[AB86] J. Archibald, J.-L. Baer, "Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model," ACM Trans. Comp. Sys, V. 4, No.4, Nov. 1986, pp. 273-298.

[ACK87] O. I. Aven, E. G. Coffman, Y. A. Kogan, "Stochastic Analysis of Computer Storage," Reidel, Amsterdam, 1987.

[AHH89] A. Agarwal, M. Horowitz, J. Hennessy, "An Analytical Cache Model," ACM Trans. Comp. Sys, V. 7, No.2, May. 1982, pp. 184-215.

[Den68] P. J. Denning, "The working Set Model for Program Behavior," Com. ACM, V.11, No. 5, May 1968, pp. 323-333.

[Fel68] W. Feller, "An Introduction to Probability Theory and Its Applications, Vol. 1," J. Wiley & Sons, 1968.

[FGT87] P. Flajolet, D. Gardy, L. Thimonier, "Birthday paradox, coupon collectors, caching algorithms and self-organizing search," Research Report RR-720, INRIA, 1987.

[FR90] C. Fricker, Ph. Robert, "On the representation of memory references generated by a program with application to the analysis of cache memories," Research Report RR-1158, INRIA, 1990.

[GMR87] A. G. Greenberg, I. Mitrani, L. Rudolph, "Analysis of Snooping Caches," Performance '87, 1987, pp. 345-361.

[Goo87] J. R. Goodman, "Cache Memory Optimization to Reduce Processor/Memory Traffic," J. VLSI and Compt. Syst., 1987, pp. 61-86.

[Len74] J. Lenfant, "Evaluation sur des Modèles de comportement de programme de la taille d'un ensemble de travail," Research Report RR-55, INRIA, 1974.

[Rao78] G. S. Rao, "Performance Analysis of Cache Memories," J. ACM, V.25, No. 3, July 1978, pp. 378-395.

[Smi82] A. J. Smith, " Cache Memories," Computing Survey, V.14, No. 3, Sept. 1982, pp. 473-530.

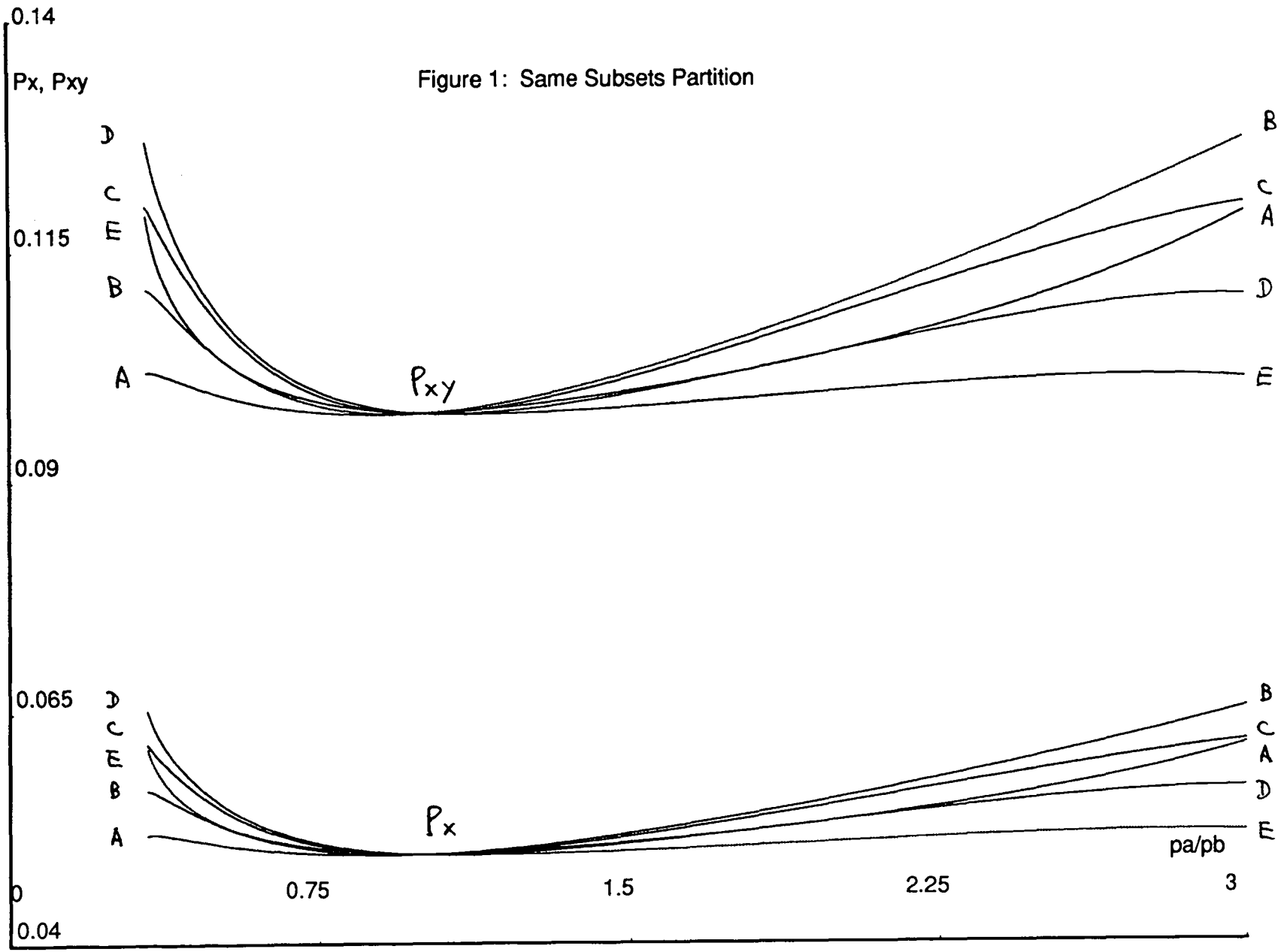

Figure 1: Same Subsets Partition

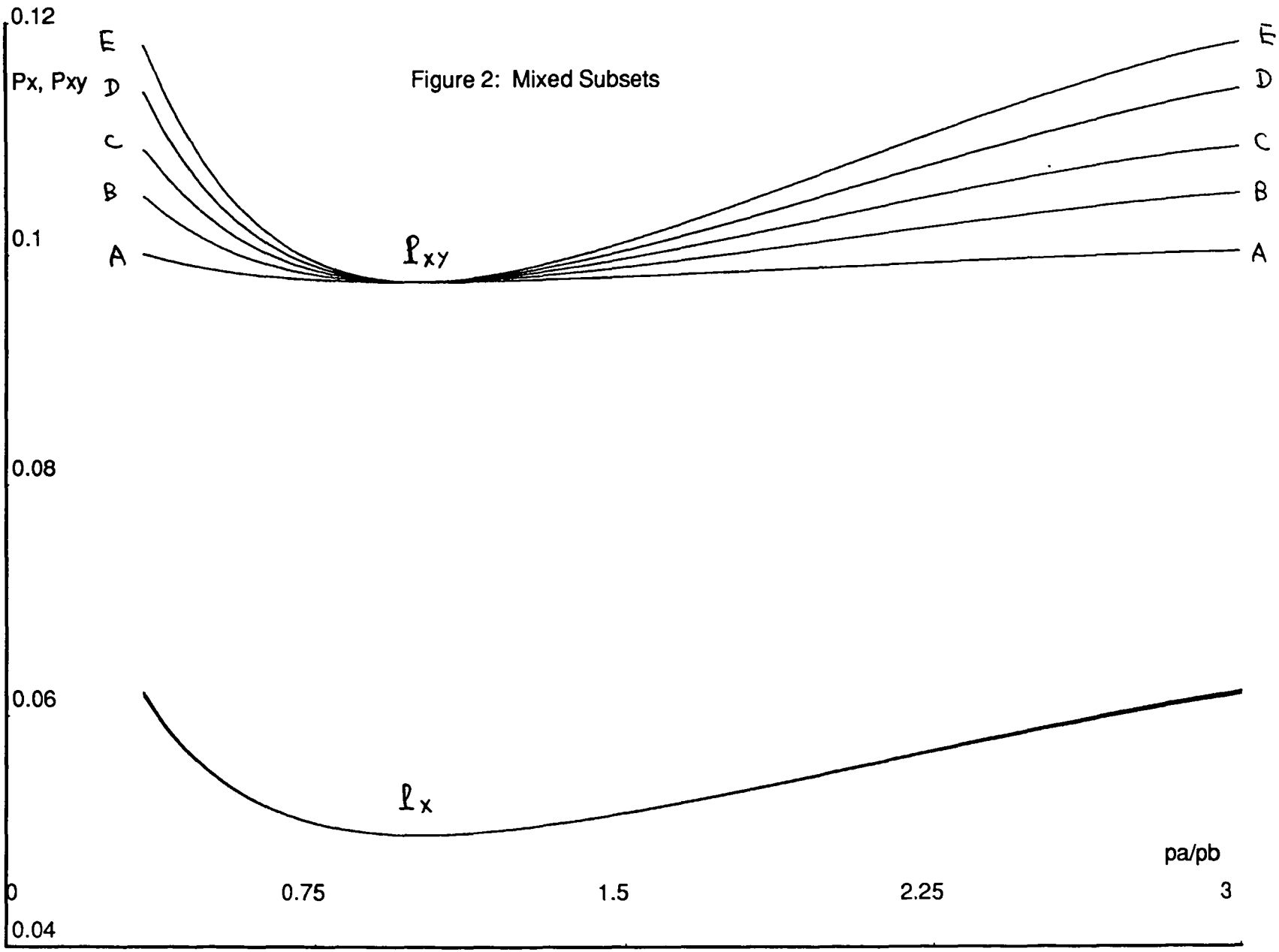

Figure 2: Mixed Subsets

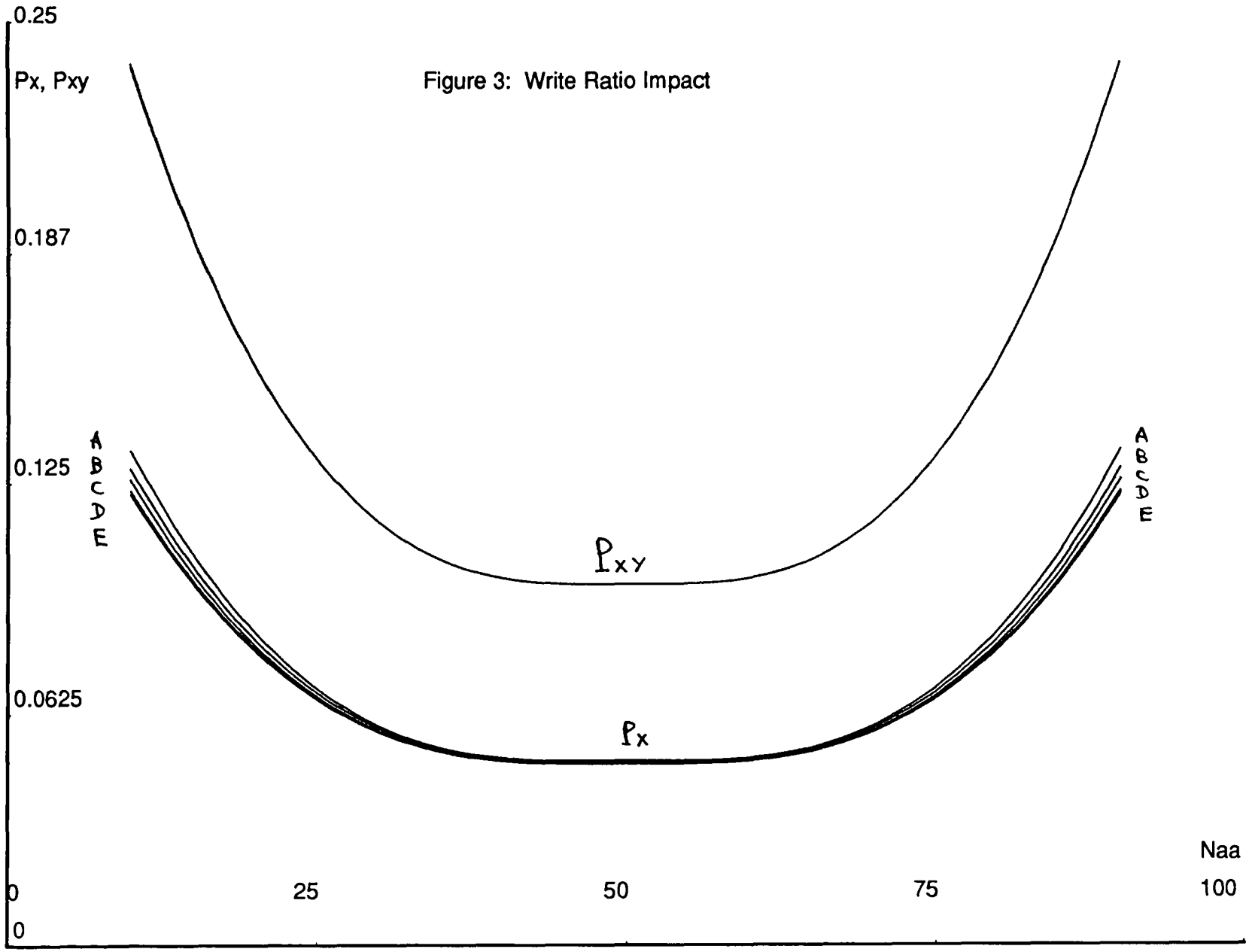

Figure 3: Write Ratio Impact

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399